

UVG

Algoritmos y Programación Básica

Rodolfo Galdámez – 11040

Alejandra Reyes-11231

Investigación corta

1. ¿Qué nos piden?
 - Elaborar un juego de BlackJack con el programa de Python.
 - Opción para dos jugadores.
 - Desplegar gráficos de resultados

Meta

- La Meta es hacer que el programa funcione como el juego de **“Black Jack”**.

Entradas

- Iniciar partida o salir
- Nombre de los jugadores
- Desea otra carta o plantear el juego
- Desea una nueva partida si/no

Salidas

- Nombre del jugador con sus respectivas cartas y la suma de ellas
- Gano o Perdió
- Gracias por jugar BlackJack
- Grafica de estadísticas

2. ¿Qué nos dan?
 - Dos cartas (números)
 - Opción de pedir una carta mas o plantear el juego (quedarme con las cartas que tengo)
 - Si los jugadores tienen la misma suma de cartas es un empate
 - Si uno de los jugadores obtiene un número mayor que el otro sin pasarse de 21 gana.

3. ¿Restricciones?

- Si un jugador obtiene más de 21 pierde automáticamente, dándole así la victoria al otro jugador.
- Solo se puede pedir una carta mas a la vez (por turno del jugador)
- Debe de ingresar lo que las instrucciones le pidan, de lo contrario le marcara ERROR.

4. ¿Repeticiones?

- Pedir una carta/ plantear juego
- Desea volver a jugar
- Nombre de los jugadores

Descripción del programa

Se desarrollo un juego con programación defensiva y cualidades como un despliegue de gráficas mostrando el desempeño de los jugadores. Se desarrollaron funciones distintas con tareas específicas.

Este juego está elaborado para que jueguen únicamente dos personas. Por medio de repeticiones con la palabra reservada `'while'`, se logra que el programa permita jugar varias rondas y elegir si queremos continuar el juego o terminar. Finalmente, al ingresar la palabra `'no'` en la opción de si desea una nueva partida, el programa crea y despliega dos gráficas: la primera (grafica de pie) demuestra en qué porcentaje de juegos ha ganado cada jugador, y la segunda (de barras) muestra las veces que cada jugador gano y/o las veces que se empato.

Diseño

Iniciamos las variables que vamos a utilizar como contadores que empiecen desde cero. Se importo el módulo `'random'`, con el cual la maquina dará aleatoriamente 2 cartas con cualquier numero y el modulo `'pylab'` para desplegar gráficos del juego.

- Funciones:

“Una función es un fragmento de código con un nombre asociado que realiza una serie de tareas y devuelve un valor. A los fragmentos de código que tienen un nombre asociado y no devuelven valores se les suele llamar procedimientos. En Python no existen los procedimientos, ya que cuando el programador no especifica un valor de retorno la función devuelve el valor `None`, equivalente al `Null` de Java. Además de ayudarnos a programar y depurar dividiendo el programa en partes las funciones también permiten reutilizar código.”

(Publicación en <http://mundogeek.net/archivos/2008/03/03/python-funciones/>)

a. Función `jug1 (total,total2):`

Esta función guarda el numero de ambas cartas de el jugador que han sido dadas por la maquina y la suma de las mismas. Pregunta al jugador si este desea una nueva carta o plantear el juego. Si plantea juego (Ingrese 2) mantiene el número inicial. Pero si el jugador desea una nueva carta (Ingrese 1) la computadora le dará una mas y sumara su puntaje. Si esta nueva suma excede el numero 21 el juego terminara automáticamente e indicara que este perdió, dándole así la victoria otro jugador. Por lo contrario si el jugador obtiene 21 este ganara automáticamente (BlackJack).

b. Función `jug2 (total,total2):`

Esta función guarda el numero de ambas cartas de el jugador que han sido dadas por la maquina y la suma de las mismas. Pregunta al jugador si este desea una nueva carta o plantear el juego. Si plantea juego (Ingrese 2) mantiene el número inicial. Pero si el jugador desea una nueva carta (Ingrese 1) la computadora le dará una mas y sumara su puntaje. Si esta nueva suma excede el numero 21 el juego terminara automáticamente e indicara que este perdió, dándole así la victoria otro jugador. Por lo contrario si el jugador obtiene 21 este ganara automáticamente (BlackJack).

c. Función `terminar(total,total2):`

En esta función por cada turno que pase de ambos jugadores se les preguntara de nuevo si quieren otro turno o terminar el juego. Si uno de ellos quiere otro turno (Ingrese 1) puede pedirlo y se les dará la opción de pedir una nueva carta (Ingrese 1) o plantear el juego (Ingrese 2) a ambos jugadores. De ser lo contrario, es decir, desean terminar el juego (Ingrese 2) se compararan los resultados de ambos jugadores y se determinará si se gano, perdió o hubo un empate.

d. Función `Seguir_pre():`

Esta función nos ayuda al momento de terminar una partida. Se les preguntara a los jugadores si desean seguir jugando. Si desea un nuevo juego (si) se creara una nueva partida. De ser la respuesta 'no', el programa desplegara un mensaje de despedida y dos graficas mostrando el rendimiento de cada jugador.

e. Función main():

Esta función tiene como objetivo desplegar la primera información del juego.

Si no desea jugar (Ingrese 1) desplegara un mensaje de despedida, de lo contrario (Ingrese 2) iniciara el juego desplegando:

- Primeras instrucciones del juego.

- Pide y guarda el nombre de ambos jugadores.

- Reparte a cada jugador dos cartas (numero) y las suma de las mismas.

También se le agregaron parámetros para que la maquina no desplegara ningún numero negativo ni mayor a diez.

Nota: En cualquiera de estas funciones, al ingresar un número o argumento inválido desplegara un mensaje de error y hará la respectiva pregunta de nuevo hasta que se ingrese un número o argumento válido, esto es por la programación defensiva que tiene el juego.

Conclusión

Aprendimos a hacer programación defensiva por medio de funciones y palabras reservadas. A crear varias funciones para un mismo programa y a utilizar repeticiones para así poder sintetizar el trabajo.

Logramos aprender a utilizar las funciones como *import* que nos ayudan a ahorrarnos varios pasos en un programa y para poder hacer el programa más atractivo desplegando, en este caso, gráficos.

Bibliografía citada:

- Publicación en <http://mundogeek.net/archivos/2008/03/03/python-funciones/>
- Libros Python en <http://sakai.uvg.edu.gt>